**Addressing modes:**

* The 8085 microprocessor supports various addressing modes that enable efficient data access and manipulation.
* These modes define how memory or registers are referenced in instructions, optimizing performance and simplifying programming.

**What is Addressing Mode?**

* An addressing mode in a microprocessor defines the way in which the operand (data) is specified for an instruction.
* It determines how the data is accessed from memory or registers and how it is operated upon.
* Different addressing modes provide flexibility, helping to optimize performance, save memory, and simplify code.

**Types of Addressing Modes:**

**1. Immediate Addressing Mode**

**Definition:**

* In this mode, the operand (data) is directly specified in the instruction.
* The data is stored in the instruction itself.
* If the data is 8-bit, the instruction takes **2 bytes**.
* If the data is 16-bit, the instruction takes **3 bytes**.

**Examples:**

* MVI B, 45H → Move the immediate value **45H** into register **B**.
* LXI H, 3050H → Load the H-L register pair with the immediate value **3050H**.
* JMP 2000H → Jump to memory address **2000H** immediately.

**Use Case:**

* Used when a fixed value needs to be assigned to a register or memory location.

**2. Register Addressing Mode**

**Definition:**

* The data is stored in a register, and operations are performed using registers.
* The instruction specifies the register(s) involved in the operation.

**Examples:**

* MOV A, B → Copy the value from register **B** into register **A**.
* ADD B → Add the value of register **B** to **A** and store the result in **A**.
* INR A → Increment the value of register **A** by **1**.

**Use Case:**

* Faster execution as operations are performed within registers, avoiding memory access.

**3. Direct Addressing Mode**

**Definition:**

* The memory address of the operand is directly specified in the instruction.
* The instruction fetches data from the specified memory location.

**Examples:**

* LDA 2050H → Load the contents of memory location **2050H** into the **accumulator (A)**.
* LHLD 3050H → Load the contents of memory location **3050H** into the **H-L register pair**.
* IN 35H → Read data from **input port 35H**.

**Use Case:**

* Useful when fetching data directly from a known memory address.

**4. Register Indirect Addressing Mode**

**Definition:**

* The operand is stored in memory, but the memory address is specified indirectly through a register pair.
* The register pair holds the memory location where the data is stored.

**Examples:**

* MOV A, M → Load the data from the memory location pointed to by **H-L pair** into the accumulator.
* LDAX B → Load the data from the memory location pointed by the **B-C register pair** into **A**.
* STAX B → Store the contents of **A** into the memory location pointed by the **B-C register pair**.

**Use Case:**

* Used when working with arrays, tables, or pointers in memory.

**5. Implied/Implicit Addressing Mode**

**Definition:**

* The operand is not explicitly mentioned; the instruction itself defines the operation and the register involved.
* The microprocessor inherently knows which register to operate on.

**Examples:**

* CMA → Complement (flip) all bits of the **accumulator (A)**.
* RRC → Rotate the bits of **A** right by one position.
* RLC → Rotate the bits of **A** left by one position.

**Use Case:**

* Used for instructions that inherently operate on specific registers (e.g., accumulator operations).

**6. Relative Addressing Mode**

**Definition:**

* The operand is a memory address determined by adding an **offset (constant value)** to the Program Counter (PC).
* This mode is used for branching (jumping) to nearby locations within the program.

**Example:**

MOV R0, #05H

AGAIN:

MVI A, #55H

ADD A, R0

JMP AGAIN

* The instruction JMP AGAIN jumps to the **AGAIN** label by adding an offset to the **Program Counter (PC)**.
* This creates a **loop** that executes continuously.

**Use Case:**

* Used in program loops and conditional jumps.